

# Assignment Name: Simple CSS

Due Date: See Canvas

## Goals:

You have been given an HTML file which has all of the required content to recreate the provided screenshot using only CSS. Do not change the body of the html file, just the head to link the CSS stylesheets and favicons. As always, use best practices, e.g. using an (unmodified) default style sheet, only lowercase letters, syntax and accessibility validation, etc.

## Objectives:

1. Demonstrate ability to write simple CSS - particularly the box model.
2. Use hexadecimal values for all colors (please use full 6 digits for autograder purposes)
3. Demonstrate understanding of cascade properties in CSS
4. Demonstrate the ability to use accessibility checkers on your code.
5. Demonstrate ability to host files properly

## Resources:

[https://github.com/UMSIComplexWebDesign/hw2\\_simple\\_css](https://github.com/UMSIComplexWebDesign/hw2_simple_css)

You are provided with a repo that contains the necessary html file, the default stylesheet, and screenshots of the final results. You will need to create a css file that will transform the original page to one that resembles our example using the outlined CSS styles below and link it to the html file.

## Requirements:

1. All files and folders present, HTML file and CSS stylesheet in proper format, no styles in HTML. **DO NOT MAKE ANY CHANGES TO THE HTML BODY –JUST THE HEAD.**
2. Always name your stylesheet **style.css** and keep it in the css folder
3. Always name the default stylesheet **html5reset.css** and keep it in the css folder (you should use the same file from homework 1 but I provided a copy this time just in case)
4. Validate all of the code - do this throughout the entire process.

5. [Utilize the Homework Submission Checklist!](#)

---

1. **Link the stylesheets correctly.**

2. **Link the favicon correctly.**

- a. `<link rel="icon" type="image/x-icon" href="/images/favicon.ico">`

3. **Style the body of the page with:**

- a. Default text is **Arial** font with **Verdana** as a fallback option
- b. Black background (use hexadecimal values)
- c. White font (use hexadecimal values)
- d. Margin of 20px on the top and sides. None on the bottom.

4. **Style the h1 heading with:**

- a. 42px centered font
- b. Padding of 15px

5. **Style the h2 heading with:**

- a. 30px dark gray font (#0D0D0D)

6. **Style the default paragraphs with:**

- a. 24px font
- b. Padding of 10px on the top and left only

7. **Style the default style for images so that:**

- a. All images are 250px wide (do not set a height)
- b. All images are centered
- c. All images have a white 10px dashed border

8. **Style the default div elements with:**

- a. A flexible height of **at least** 125px tall
- b. Padding of 20px on all sides
- c. A 3px solid white border on all sides.

**9. Style the introduction class with:**

- a. 15px border-radius (to curve the edges)
- b. Margin of 15px margin on the top
- c. Double-spaced text

**10. Style the red class with:**

- a. Background color is hexadecimal value for red
- b. Text is aligned on the right side
- c. A left margin of 50px;
- d. A border of 2px on the bottom and 5px on the left;

**11. The blue div has the following styles**

- a. Background color is hex value for blue
- b. Text is aligned on the left side
- c. A right margin of 50px;
- d. A border of 5px on the right and 2px on the top;

WAVE YOUR PAGE!!!! Uh oh. *Let's fix the color contrast here – Dark on blue is too hard to read.*

- e. Use #C4C4C4 for the heading and paragraph text

**12. The yellow div has the following styles**

- a. Background color is hexadecimal value for yellow
- b. Text is aligned on the right side
- c. A left margin of 50px;
- d. A border of 2px on the bottom and 5px on the left;

WAVE YOUR PAGE!!!! Uh oh. *Let's fix the color contrast here too*

- e. Use #707070 for the paragraph text

**13. The green div has the following styles:**

- a. The background color is a hexadecimal value for green (You must use #00EB00, #008500, **or** #00D800)
- b. Text is aligned on the left side

- c. A right margin of 50px;
- d. A border of 5px on the right and 2px on the top;

**14. Style the default footer elements with:**

- a. a [gradient background](#) that starts at #000000, goes to #0000ff at 25, #0066ff at 70, and ends back at #000000.
- b. A 5px solid white border.

**15. Style any paragraphs within the footer with:**

- a. 20px text that is aligned to the right.
- b. Padding of 15px on the top, right, and bottom only. (No padding on the left.)

## Example

See the Github repo for a sample screenshot at 1200px. (Here is a link to the [HW2 screenshot image](#) as well.)

*I noticed that some people have an extension that is adding an extra div to the bottom of your page. Check the page on a different browser if that is the case for you.*

## Tips!

1. Write your CSS from broadest to narrowest. Look at the requirements, which ones apply to the whole page? Which ones only apply to certain sections? Use this to help define your CSS selectors.
2. If you get an error, use Inspect Element to check the current value. **Make sure that a different rule isn't overriding a rule you wrote.**
3. Don't forget that you can validate CSS code: <https://jigsaw.w3.org/css-validator/> is one.
4. Order is important for many CSS parameters, even your fonts! Make sure your desired font and the fallback are in the right order.
5. Images can be a little tricky, what CSS parameter can we use to make sure it's always on its own line?
6. Automatically centering elements seems hard but you really only need 1 line of code! Do you remember what it is?
7. CSS means cascading style sheets, use the cascade to your advantage. Sometimes you may only need 1 style for a selector.

## Submission Requirements:

1. Upload your code to Github in a **repo called hw2-w25**. Make sure to make this into a Github Page so we can see your deployed code.
2. Submit link to the autograder in Canvas. We want to see the deployed page, not the code so the URL should be **`http://<your-user-name>.github.io/hw2-w25`**
3. **If you are getting a URL error in the Autograder please let us know!**

Any changes to code after the due date can result in a late penalty at any time during the semester. If you want to modify your code for personal use - make a new copy.

## Autograder

1. If you are losing points on the autograder, use Inspect Element to carefully check that the element has the styling that you were expecting.
2. If the autograder is giving you a 0
  - a. Check the URL you submitted, open it in your own browser, does it go to the proper page?
  - b. Validate your CSS to make sure you don't have any syntax errors.

## Deeper Questions...

1. What error would be generated if you made the paragraph text smaller?
2. What visual differences are there if the html5reset file is not used?
3. Are any of your rules repeated in multiple places in the css? Could you do more to optimize your code?